CTCV: A Protocol for Coordinated Transport of Correlated Video in Smart Camera Networks

Vinay Kolar*, Israat Tanzeena Haque†, Vikram P. Munishwar*, Nael B. Abu-Ghazaleh†
*Cisco Systems, San Jose
{vinkolar,vmunishw}@cisco.com
† Dept. of Computer Science and Engineering, University of California, Riverside
{israat,nael}@ucr.edu

***Abstract*—Smart camera networks (SCNs) are increasingly used in applications such as homeland security, border control and traffic monitoring. The cameras are often wireless with an organically growing structure, to reduce the overhead of deployment. We frame a new and important problem in SCNs: how to transmit videos from multiple cameras with *overlapping coverage* given the limited available wireless bandwidth to maximize the quality of the received videos. We call this problem *Coordinated Transport of Correlated Videos* (CTCV). CTCV is a more general version of 3D video transport: in that problem, highly correlated videos from two cameras (that provide the 3D perspective) are jointly encoded exploiting their pre-defined and known overlap. In contrast, in CTCV there is an arbitrary number of cameras whose overlap is not known apriori and that require transmission as multiple video streams. To effectively support CTCV, we propose a video delivery protocol that consists of two primary components: (1) Consolidation of correlated videos from multiple cameras which removes spatially redundant fields-of-view; and (2) Network and coverage aware bandwidth allocation to optimize coverage quality cooperatively among the different video streams to match the available bandwidth. We formulate the problem of optimal bandwidth allocation for maximizing coverage. We propose and investigate different heuristic policies for bandwidth allocation. We evaluate CTCV using data from a small camera testbed as well as topologies from realistic deployments. Experiments show that CTCV achieves around 10 dB gains in video quality in the scenarios we consider.**

I. Introduction

Recent developments in network cameras and wireless technology have enabled deployments of Smart Camera Networks (SCNs) in many application domains that benefit from visual surveillance. SCNs collaboratively self-configure, adapting to what they are monitoring to improve the network's overall effectiveness, and to lower reliance on human operators [1]. SCNs often are deployed with limited planning using wireless networking and evolve *organically* over years. For example, the Chicago video surveillance system, started around 2002, grew from a test deployment to 8000 cameras over its first five years, and now has an estimated 20,000 cameras. This surveillance system grew organically when multiple organizations and companies – over a period of time – contributed their camera feeds for public surveillance [2].

In an SCN, monitoring of an event (either in real-time or after the fact) requires retrieval and presentation of multiple video streams from a group of nearby cameras that are spatially correlated (include overlap in coverage). Since video consumes significant bandwidth, it is easy to overwhelm the limited bandwidth of the network, especially in wireless deployments, leading to poor-quality delivered video [3], [4]. We call this problem *Coordinated Transport of Correlated Videos (CTCV)*. The correlated nature of the videos opens up opportunities to reduce redundancy and therefore the amount of data that needs to be transported. In particular, the streams are typically combined to present the viewer a unified intuitive view. Finally, as congestion occurs, the video quality must be reduced to match the available bandwidth. However, it is necessary to carry out such decisions in a way that maintains coverage and delivers acceptable video quality.

To effectively address the problem above, we propose a coordinated video transport protocol. The protocol leverages two primary mechanisms:

- **Video combination and redundancy removal**: In a vast majority of SCN deployments, the video feeds from individual cameras are combined at the Observation Center (OC) to present a single view to the observers for ease of monitoring. CTCV combines videos within network to eliminate redundancy originating from the overlapping fields-of-view (FoVs) of cameras. We also investigate detecting the overlap and informing the cameras to eliminate the redundancy at the source. Eliminating redundancy from data is similar to aggregation in conventional sensor networks [5]–[8]. However, detecting overlaps and merging videos from different view points is significantly more complicated. To our knowledge, this is the first time in-network aggregation is applied to video data.
- **Coverage aware bandwidth allocation**: Having reduced the size of video by eliminating redundancy, the second component of CTCV controls the data rates of the individual and combined videos to ensure that they fit within the available network bandwidth. The wireless channel is shared with nearby cameras and routers; the decision on bandwidth allocation must be coordinated. At the same time, the value of video streams being transmitted may be different (as measured by the coverage area, or other application specific metrics). This is a Generalized Network Utility Maximization (GNUM) problem [9];

we formulate a coverage-aware joint routing and rate control problem. To provide low-overhead solutions, and to account for effects present in wireless networks such as external interference and CSMA scheduling, we introduce several heuristics to solve the problem while maintaining coverage-level fairness. While there is significant work in wireless multimedia networking on adapting video rate to available bandwidth, the vast majority of it targets a single video feed/camera [10]–[12].

We evaluate the proposed ideas both in a small multi-camera testbed as well as using simulation. For simulation, we model two SCNs based on recent deployments and using publicly available video feeds. One scenario is modeled after a highway monitoring SCN, while the other models a border-monitoring deployment. Since video processing operations such as the ones proposed here are difficult to evaluate accurately in network simulators that statistically model video generation, we modify the simulator to operate on pre-collected videos that are played back and processed during simulation. This methodology allows us to accurately evaluate the impact of networking and video events on the overall delivered video quality. Our results show that CTCV continues to deliver high quality video – with Peak Signal-to-Noise Ratio (PSNR) video quality [13] of 33 dB – when the standard protocols deliver unviewable video quality (PSNR of 22 dB) due to large number of video frame errors.

In summary, the contributions of the paper are:

1) We define the new problem of Coordinated Transport of Correlated Videos that arises in the context of Smart Camera Networks. We propose a solution to this problem consisting of video consolidation as well as bandwidth and coverage aware rate allocation. We model the problem as a coverage-aware Network Utility Maximization problem, and propose heuristics for CTCV that accounts for practical network effects such as CSMA scheduling.
2) We propose video consolidation to remove redundant overlap regions from videos being transported. We develop a video stitching implementation that is 15x faster than image by image stitching. We investigated consolidation in the network, as well as at the end cameras. To our knowledge, this is the first proposal for sensor network aggregation on video data.
3) We propose a coordinated coverage and network aware bandwidth allocation and video rate control. We express the problem formally and develop solutions for it. To our knowledge, this is the first video delivery model targeted towards multimedia networking of multiple related video streams.
4) We investigate CTCV both in a camera testbed as well as using simulation, showing that it can dramatically improve performance.

II. Overview and Assumptions

We propose a Coordinated Transport of Correlated Videos (CTCV) protocol: a coverage and bandwidth-aware video transport protocol for coordinated videos. We first describe the alternatives for consolidation and the protocol (in Section III) that efficiently constructs and delivers a consolidated video of the event being monitored, while removing redundancy. We then model the problem of coverage-aware joint routing and scheduling as a Generalized Network Utility Maximization (GNUM) optimization problem (Section IV). We then propose heuristic CTCV solutions to (described in Section V) allocate the bandwidth to the streams while taking into account the coverage of the different video feeds.

We assume that the coordinated videos together serve a query at the observation center. The videos are fused into one presentation that provides a combined view of the videos to a human observer. In this paper, we use a *mosaicing* based video presentation, which stitches multiple videos from nearby regions to form a single mosaic of the covered area [14]. While mosaicing is the most common multiple-video presentation approach, other approaches such as video summarization [15] and 3D holograms [16] are possible, and would invite different in-network video combining functions. The presentation model may be adaptable/queriable by the observer, requiring a corresponding adaptation of the set of streams being forwarded. We do not consider other presentation models in the current study; this is a part of our future work.

III. Video Consolidation to remove redundancy

The first component of CTCV is video consolidation to remove redundancy. Consolidation is a form of aggregation in sensor networks for video data; we believe that this is the first implementation of aggregation in the context of a wireless sensor network for such a complex data type. We describe two flavors of consolidation, and follow by describing the design and implementation of our consolidation framework.

A. Consolidation Schemes

We propose two alternatives: (1) *Consolidation by Aggregation (CA)*, which consolidates videos at intermediate routers to remove redundancy, and (2) *Consolidation by Coordination (CC)* which pushes the consolidation to the cameras to remove redundancy at the source.

1. Consolidation by Aggregation (CA): CA performs in-network processing at the routers to remove redundancy in videos before transmitting towards the OC. Figure 1 illustrates CA. Cameras (C1-C4) view different parts of the surveillance area with possibly overlapping Fields-of-View (FoVs). The video streams from the cameras are forwarded towards the Observation Center (OC) by the video routers.

The routers decode the video from multiple video streams, and attempt to consolidate into a single video stream by removing the overlaps. The output video stream contains the stitched mosaic video from the input streams. In the Figure 1 scenario, the video streams from different cameras are decoded and then stitched into a single mosaic at Level-1 routers (R11, R12). If an FoV of a camera is completely redundant, such as C4's FoV, then its video can be eliminated. Multi-level

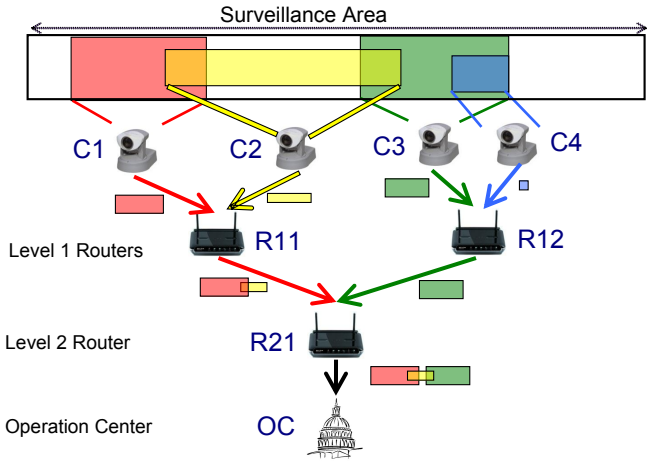


Fig. 1. Overview of Consolidation by Aggregation (CA)

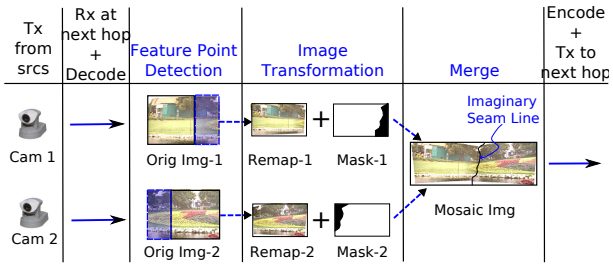


Fig. 2. Creating a Mosaic Image

consolidation is enabled by consolidating at multiple points as the video stream travels towards the OC.

2. Consolidation by Coordination (CC): CC applies in SCNs that use computationally capable cameras. In this case, the network coordinates to provide the cameras with information about redundant regions, allowing them to prune these regions without transmission. This approach removes the need for in-network processing at the routers.

B. Design and Implementation

Both our consolidation implementations assume that video mosaicing is used as a presentation model to the observer. However, both approaches in principle generalize to other presentation models assuming that redundancy can be detected and removed. Video Mosaicing is performed by repeatedly stitching image frames from the cameras which cover neighboring Fields-of-view (FoVs).

We create a mosaic image from multiple overlapping images by using existing panorama stitching tools [14]. The main stages (as shown in Figure 2) are described below.

a. Detecting the feature points between the images: Informative common points (called *feature points*) are first detected in the overlapped parts of the image. The two images are stitched by aligning the respective feature points.

b. Finding the image transformation matrix: The images at the camera are captured at different positions and camera parameters (e.g., focal length of the lens). However, the final mosaic image should fit each of the images such that it appears to be taken from a single point. Hence, each image is *remapped* such that the projection of the images fit the adjacent images. Remapping is done in two stages: calculating the *transformation matrix* to find out the transformation, and the actual transformation of the image. The transformation matrix consists of:

(1) the information to derive the remapped image from the original image (*Remap-n* in Figure 2), and

(2) the X- and Y-offset of the remapped image in the final mosaic image.

In addition, this phase creates the mask for the image; this is the region of the image that is present in the final mosaic image (*Mask-n* in Figure 2). Note that the mask may be scene-specific. If one camera captures the scene better than the other cameras (say, because of the perspective), the mask of that specific camera can include that scene.

c. Merging the images: The transformation is a simple geometric transformation of the remapped image shifted the image according the above offsets. The transformed images are then superimposed. Advanced blending algorithms can be used to smooth the seam line that appears at the boundary of the two images [17].

C. Video Consolidation using Aggregation (CA)

Similar to aggregation in sensor networks, CA attempts to consolidate data by applying the video mosaicing process (Figure 2) at every intermediate router for every video frame that is received at the router. However, CA requires synchronization of the streams. Also, CA is computationally expensive requiring the use of specialized accelerators (e.g., GPUs) to be able to scale to video rates. In contrast, CC eliminates redundancy at the cameras and removes the stitching computational burden from the routers.

We use a two phase system that efficiently stitches the video at the intermediate routers. The motivation is to reduce the overhead of computationally intensive steps – Feature point detection, transformation-matrix computation and mask generation – in mosaicing. Coincidentally, these expensive components are not required to be executed at every frame if the camera's FoV parameters do not change. Thus, we preprocess these steps once in a *Mosaicing parameter generation* phase, and use the resulting parameters repeatedly to stitch images from every frame. The combination of these techniques provides a speedup of up to 15x in our system over the standard implementation.

1. Mosaicing parameter generation: This phase is executed at a router for the first frame received from the camera or another downstream router, or when the FoV of one or more of the cameras is determined to have changed. Using the images from the frame, the system computes the mosaicing parameters at every router. This phase is essentially the control plane operation for consolidation. Whenever the FoVs change due to, say, camera movement, this phase should be re-executed to identify the new mosaicing parameters.

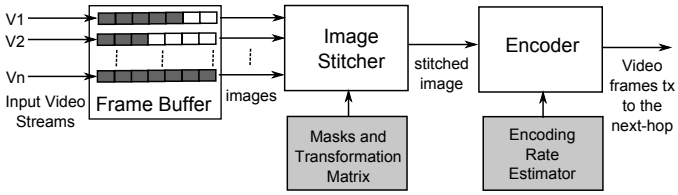


Fig. 3. Components of *Video Fusion* phase

2. Video Fusion: This phase is executed at every packet reception on the router. The router decodes the video frames, merges the frames into a mosaic, and, finally, encodes and transmits the mosaic frame to the next-hop router (as shown in Figure 3).

The input video streams to the routers consist of encoded frame data. The data for one frame can be of arbitrary size, and hence can be made up of multiple packets or fragments. To track fragments of the frame, we transmit the video ID, sequence number, frame number, frame size and fragment number in the header of each video data packet. The cameras are time-synchronized with a tolerance matching the dynamics of the video (typically within milliseconds). This is a typical requirement on multi-camera systems when the streams are being monitored together.

A *Frame Buffer* stores the packets until they are decoded into a single frame. Once the complete frame is received, it is decoded into an image, and stored in an image buffer (currently, in the YUV format). The router waits for frames from all its input streams to be available before merging. Once the frames are available, the router merges the images, encodes the image and transmits to the up-link router.

We handle complete and partial frame failures by observing the sequence number of the frames. If an aggregation point receives all frames from n but not from sequence $n-1$, then it concludes that the previous frame is not fully received. For all the correctly received frames, it creates an image using the YUV information stored in its buffer. For the partially received frames, it decodes the video frame by packing null values for the remaining bytes.

Once all the frames from the incoming videos are received, we create a new frame for the output video by: (1) transforming the image using pre-calculated mosaicing parameters, and (2) merging the transformed images.

Instead of a router, a computationally powerful middle-box or virtual network functions can be used for Video Fusion. However, for the sake of simplicity, we refer to these middle-boxes as routers in the rest of the paper.

D. Video Consolidation by Coordination (CC)

While a distributed implementation is possible for video consolidation in CC, we use the OC to periodically stitch the video frames to form one mosaic video, and in the process compute the transformation matrices and masks for each camera. This information is delivered to the respective cameras. Thus, the redundant regions of the camera FoVs are suppressed at the camera itself. Finally, the OC uses mosaicing to combine the frames from all the cameras. CC removes in-network processing completely from the intermediate routers, enabling CTCV to be implemented using standard routers. The expensive mosaicing operations are also deferred to the centralized OC which can be provisioned to have sufficient resources to carry out the mosaicing at video rate.

IV. Coverage-aware Bandwidth Allocation Model

The second component of CTCV is coverage aware bandwidth allocation. To explore the fundamental structure of this problem, we frame the coverage-aware video transport problem based on the Generalized Network Utility Maximization (GNUM) framework [9]. The problem jointly optimizes bandwidth allocation and routing based on the coverage of the cameras.

Let N be a set of nodes (cameras, routers and sink), and L be the set of links in the SCN. Let $C = \{N_1, N_2, \ldots, N_c\} \subset N$ be the set of cameras, which are the sources for video streams. Let K represent the set of connections between the cameras and their destinations (the sink). The source camera, destination and the rate of transmission for a connection k are denoted by $src(k)$, $dest(k)$ and r_k, respectively. The traffic flow allocated to link (a, b) for a connection k is denoted by $f_{k,ab}$. Let $\mathbf{f}$ denote the vector of all flows on all links.

Utility of the video stream: We express the utility of a video stream k by a convex utility function $U_k(k, r_k)$. The desired characteristics of the utility function $U_k(k, r_k)$ for the CTCV problem are: (1) The rates should be proportional to the coverage, and (2) The rates should be allocated according to *proportional fairness* since the OC seeks to view the entire region, irrespective of how small the coverage is for a camera. Without loss of generality, we consider coverage to be the size of the covered area.

The utility for a connection k is a convex function $U_k(k, r_k)$ which we formulate as follows. Let $\widehat{r_k}$ be the video rate per unit-area covered for a camera. Let the camera k transmit a region of area P_k. For simplicity, we assume that each camera transmits at-least one unit-area ($P_k \geq 1$). Then, the overall rate for the camera k is $r_k = P_k \widehat{r_k}$. To promote proportional fairness, the overall utility function for the camera is given by $U(k, r_k) = \sum_{\forall P_k} \log(\widehat{r_k}) = P_k \log r_k - P_k \log P_k$. The overall network utility is given by

$$U = \sum_{k \in K} U(k, r_k) = \left(\sum_{k \in K} P_k \log r_k \right) - Z. \qquad (1)$$

where $Z = \left(\sum_{k \in K} P_k \log P_k \right)$ is a constant for the scenario. Clearly, the function $U(k, r_k)$ is convex. The overall utility function U is also convex since $P_k \geq 1$.

The utility function can be specialized to the different consolidation schemes: Naïve, CC and CA. Let $\widehat{\mathcal{P}_k}$ be the set of unit-areas covered by the camera. For each scheme we define the set of unit-areas actually transmitted as $\mathcal{P}_k \subseteq \widehat{\mathcal{P}_k}$. Let $\widehat{P_k}$ and P_k be the number of unit-areas in the above sets, respectively.

1. Naïve scheme In this scheme, the cameras, routers and sink are incognizant of the coverage overlap: each camera transmits the entire covered area to the sink, i.e. $\mathcal{P}_k = \widehat{\mathcal{P}_k}$ and $P_k = \widehat{P_k}$.
2. CC scheme In CC, the sink communicates the useful FoV through the mask: each camera transmits only the useful area at the sink, i.e. $\mathcal{P}_k \subseteq \widehat{\mathcal{P}_k}$ and $P_k \leq \widehat{P_k}$.
3. CA scheme Here, the routers may aggregate multiple flows to form a new stream by removing overlap. The video stream is not transported end-to-end from the cameras to the sink. Instead, each video stream from the cameras is terminated at a downstream router that aggregates multiple flows, and the router will transmit a new aggregated video flow to the downstream nodes.

We split the video flow from the cameras or intermediate aggregating routers as separate flows. The connection set K consists of all end-to-end connections to the respective destinations. The number of unit-areas transmitted from the sources is equal to the the area covered, i.e. $P_k = \widehat{P_k}, \forall k \in K$ and $src(k) \in C$.

The connections at the aggregating routers will remove the overlaps and transmit a subset of the received FoV to its downstream routers. Hence, if a router N_k receives flows from routers N_a, N_b, N_c, then the new flow will have $P_k \leq \sum_{i \in \{a,b,c\}} P_i$.

All the schemes compute the P_k based on the above rules. For each scheme, we then compute the perceived utility of a flow based on Eq 1.

Routing and Scheduling Model: We use standard wireless routing and scheduling models, and maximize the CTCV-specific utility in Eq 1 for GNUM problem [18]. The optimization problem allocates feasible bandwidth on various links such that the perceived utilities of the video streams are maximized. Formally, the overall problem is defined as

$$\text{Maximize} \sum_{k \in K} P_k \log r_k \tag{2}$$

such that

$$\mathbf{f} \geq 0 \ , \tag{3}$$

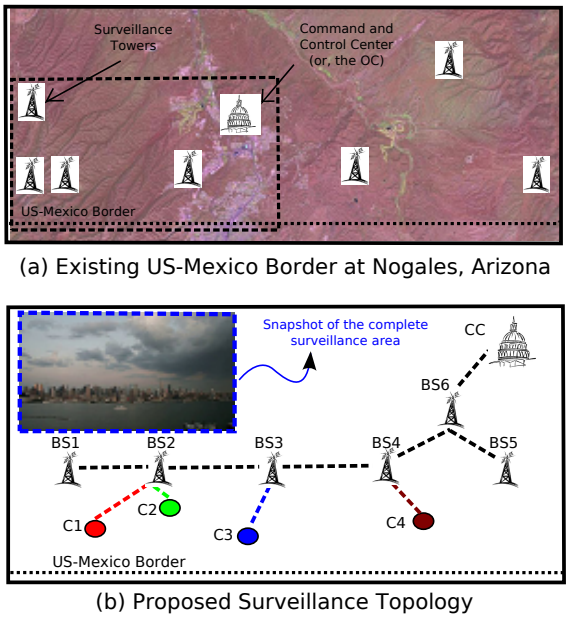
$$x_a^k \leq \sum_{b:(a,b)\in L} f_{k,ab} - \sum_{b:(b,a)\in L} f_{k,ba},$$
$$\forall k \in K, \forall a \in N, a \neq dest(k), \tag{4}$$

$$\mathbf{f} \in \Pi. \tag{5}$$

The objective function in Eq 2 is the sum of the utilities of all the video streams. The routing constraint (Eq 4) states the well-known mass balance constraints which is used to route the packets from the source to the destination.

Finally, the scheduling constraint is given by Eq 5, which defines the feasibility of the links schedule after accounting for the wireless interference. Based on the interference model used, we can define a scheduling region Π. We utilize a standard model for computing schedules [18]. Here, Π is derived by first computing the *Conflict Graph* for the wireless network which denotes the mapping between the pair of links which interfere with each other. Based on the conflict graph, feasible schedules are then created [18].

For the numerical evaluation in this paper, we model the conflict graph based on *Primary Interference Model* [18], where links that contain a common node cannot schedule together. Although this model has limitations, it has been extensively used in other GNUM studies due to its simplicity.



(a) Existing US-Mexico Border at Nogales, Arizona

(b) Proposed Surveillance Topology

Fig. 4. Surveillance topology at Nogales, Arizona: Figure (a) shows the existing terrain and infrastructure. A part of the area (inside the blue-dashed line) is considered in Figure (b). We propose real-time video streaming from cameras (C1-C4) to the control center(CC) using WiMAX base stations (BS1–BS6).

V. Coverage Aware Coordinated Bandwidth Allocation–Protocol Design

In this section, we pursue practical distributed protocols for bandwidth allocation at intermediate routers under CA. Bandwidth allocation is the problem of deciding how to allocate the shared bandwidth between nearby interfering routers. Since the videos can represent different numbers of sources and different aggregate coverage, uniform bandwidth allocation often results in sub-optimal operation.

We motivate the problem using two scenarios. Consider the scenario in Figure 1, which has a symmetric structure; each router has an equal number of streams coming from cameras or other routers. Bandwidth allocation in such scenarios may be straightforward if the coverage of the cameras is of equal value. The second example is based on the proposed wide-area border surveillance system in Arizona as shown in Figure 4 (b). Some cameras are a few hops away from the OC, and are therefore able to obtain larger throughput (e.g., C4), while other farther cameras can expect significantly lower throughput (e.g., C1 and C2). The coverage value of different cameras

differs based on overlap and the importance of the area they are monitoring.

We propose three algorithms for bandwidth allocation. They require communication between nearby routers to exchange relevant information and to agree on the allocation decisions.

1. Hierarchical Point Fair (HP-Fair) Allocation: SCNs in which cameras are connected to the OC using hierarchical network topologies (clustered topologies, cellular or WiFi-based) can be aggregated at their Hierarchical aggregation Points (HP) such as cluster-heads and base-stations. The HP-Fair algorithm assigns a single rate to each link irrespective of the number of video streams passing through it.

HP-Fair can be unfair since it assigns a single rate to all incoming videos irrespective of the contents of the video. For example, if one incoming video has a larger coverage region than another, HP-Fair assigns equal rate to both.

2. Cam-Fair aggregation: Cam-Fair improves HP-Fair by explicitly assigning bandwidth such that the encoding bit-rate from each camera is identical. Thus, each stream is allocated bandwidth proportional to the number of cameras it aggregates. For example, for the topology in Figure 4 (b), if the bit-rate per camera is 1 Mbps, then BS2 consolidates its mosaic at 2 Mbps, and BS3 at 3 Mbps, and the final mosaic at BS4 is assigned 4 Mbps.

While Cam-Fair accounts for stream coverage, it does not directly account for the final area covered at the OC. This is because Cam-Fair does not consider the part of the FoV of the camera video that finally makes it to the mosaic video at the OC. For example, in Figure 4 (b), cameras C1, C2 and C3 are closer to each other, and hence have a large overlap in their FoVs. However, camera C4 has a less overlap with other cameras, and contributes more coverage value to the final mosaic and should be allocated higher bandwidth.

3. Coverage Fair (Cov-Fair) aggregation: Cov-Fair aggregation assigns a bit-rate for each camera FoV that is proportional to the contribution of the camera FoV to the final mosaic. This is performed by first estimating the bit-rate of the video that the OC can receive. OC measures the available network bandwidth, and sets a single bit-rate (B) for the final mosaic video. If the total area covered at the OC is A, and camera C contributes an area a_C, then the video stream is assigned a bit-rate of:

$$\mathrm{Rate(C)} = \frac{Ba_C}{A}.$$

The bit-rate at each HP is set to the sum of $\mathrm{Rate(C)}$, for all the video stream from cameras C that pass through the HP.

Bandwidth Allocation in CC: Recall that, in CC, each camera is aware of the the non-overlapping area covered since the OC transmits the masks of the FoV that the cameras have to transmit. Using this information and the available bandwidth, the cameras encode each video with a rate that is computed similar to Cov-Fair scheme in CA. The available bandwidth information can be obtained, for example, as back-pressure feedback from intermediate routers, or estimated end to end. We conjecture that the optimal protocol for bandwidth allocation can be derived from the decomposition of the primal problem in Eq 2.

VI. Performance Evaluation

In this section, we evaluate CTCV bandwidth allocation problem and the heuristic protocol.

A. Analysis of optimal bandwidth allocation

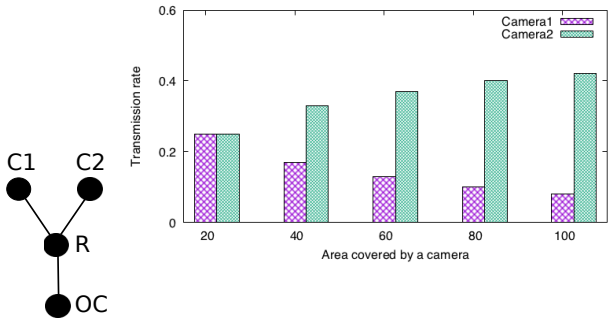


Fig. 5. Topology and bandwidth allocation

We first evaluate the model for a scenario of two cameras with two flows and a single OC (Figure 5). We keep the coverage of one camera fixed while vary the coverage for the other one. Figure 5 shows the rates achieved by each camera is linear proportional as we vary the overlap between the cameras.

Next we consider the effectiveness of naïve, CC, and CA schemes in a tree topology presented in Figure 1.We vary the percentage of the overlap between the consecutive cameras from 0 to 100. We measure the network bandwidth efficiency under the optimal rate allocation scheme: the ratio of the actual network bandwidth consumed (across all the links) to transmit the unit-areas which are present in the final mosaic video to the total bandwidth consumed.

Figure 6 shows the network efficiency. The network efficiency of CC (Coordination) is clearly 1 since all the bandwidth is used to transmit non-overlapping parts of the image. The naïve scheme degrades fast as the amount of overlap increases. CA (Aggregation) performs better than the naïve scheme since it removes the overlaps at the intermediate routers R11, R12 and R21.

B. Evaluation of Heuristic Protocols

We study various types of surveillance topologies including a small multi-camera testbed, as well as scenarios inspired by existing SCNs that use WiFi and WiMAX. We evaluate realistic CA and CC using videos collected from our camera testbed as well as public surveillance videos.

We implemented CTCV with in-network Mosaicing in the Qualnet simulator [19]. In particular, the cameras in our Qualnet model are made to send the video data provided to them as input traces. Multi-level encoding and decoding

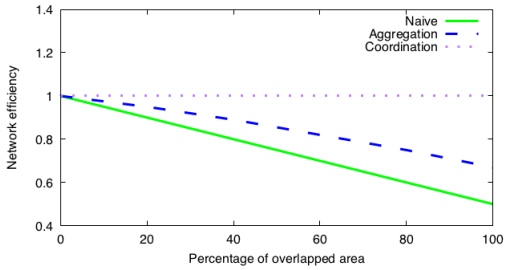


Fig. 6. The network efficiency for different streaming schemes.

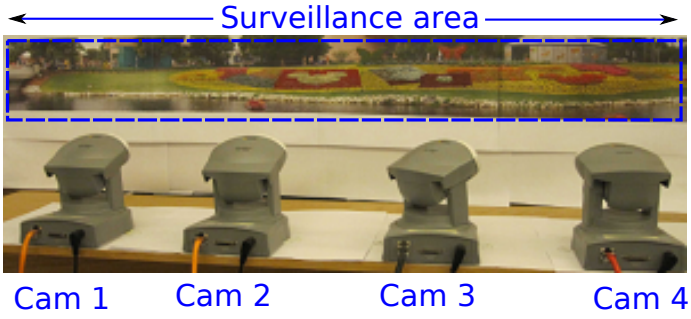


Fig. 7. Miniature Camera Testbed

is sensitive to frame packet losses since packet drops at one hop have a cascading effect on mosaic videos at later hops. Therefore, existing VBR or trace-based simulation is not sufficient [20]. We altered the simulator to provide simulation-time in-network encoding and decoding of video streams using the *ffmpeg* library [21]. Thus, all video processing is executed completely, and network effects are simulated by Qualnet.

We first evaluate a simple two-level hierarchical topology using videos generated from our testbed cameras. We then study a large symmetric topology. Finally, we analyze CA and CC in asymmetric topologies where bandwidth allocation strategies have a significant effect on the performance.

We use a small camera testbed consisting of four Axis 213 network cameras viewing different parts of a panoramic image (Figure 7). We set the cameras to have an approximate overlap of 20% in their FoVs. Each camera streams a high-resolution 4 Mbps MPEG encoded video. These videos are collected and then replayed in the simulator environment. This approach aids in validating realistic stitching from multiple cameras that are viewing different parts of the scene, while using real videos from a small testbed.

The testbed scenario is similar to Figure 1. We use constant-rate 54 Mbps IEEE 802.11a protocol with standard parameters. Video quality is measured using the standard Peak Signal-to-Noise Ratio (PSNR) of the video frames [13]: higher PSNR indicates better video quality. Usually, a PSNR of above 30 is considered good for compressed MPEG videos.

We compare CA with *standard routing* which is unaware of coverage, nature of video traffic or network capacity. Figure 8(a) shows the PSNR and frame success rate in different schemes. The standard routing scheme when transmitting videos at rates greater than 3-4 Mbps results in many frames with errors, which drastically reduces the video quality.

This behavior is exacerbated by the bursty nature of the video transmission; we noticed a peak-to-average ratio of bit-rate of above 10. Video encoding consists of two types of video frames: key frames and non-key frames. In general, key frames are much larger than other frames and are sent about once a second. In this scenario with 400 kbps videos, the average key frame size is around 25 Kbytes, where as the non-key frames are 1500 bytes.

In addition, we also measure the receiver throughput, good-put (bit-rate of frames that are received without errors) and energy consumption as shown in Figure 8 (b)(c). Clearly, goodput and frame losses are better indicators of video quality than throughput. Note that CA removes the overlapped areas at routers and provides control to regulate downstream traffic near the sink, where congestion is most likely. This enables CA to adjust bandwidth to provide better video quality.

CA also leads to energy savings (Figure 8(c)). Transmission near the ideal aggregation rate saves energy by around 13%, in addition to improving quality. The energy saving occur because of the lower transmission rates and retransmissions.

In the next experiment, we simulate surveillance of a part of Highway-A7 in Germany (Figure 9). The highway already has surveillance camera infrastructure where cameras are placed at half a kilometer from each other. This SCN has a large but symmetric topology.

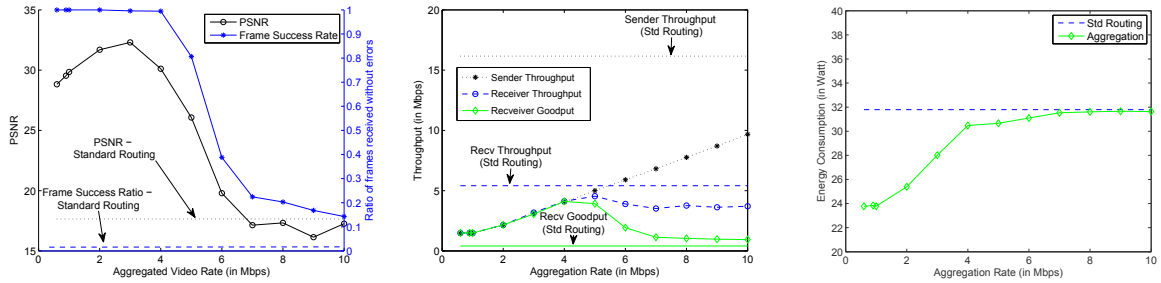
We extrapolate the environment with additional cameras deployed to improve the coverage. We assume that the cameras are placed every 50 m (say, on light poles). The inset figure shows the proposed camera placement over a stretch of 500 m. We evaluate a hierarchical topology to deliver the videos from these cameras. Four consecutive cameras are attached to a router which transmits to the relay node.

In addition to PSNR, we measure the video quality using the Structural Similarity (SSIM) [22] metric. SSIM measures quality of decoded video by comparing the structure of original image with the decoded video – instead of comparing pixel values (as done in PSNR). SSIM is a value in the range $[-1, 1]$, with higher values corresponding to better quality. SSIM evaluates the video quality as perceived by the human eye, and is therefore more accurate than an pixel-level metric such as PSNR [22].

Figure 10 shows the quality of received videos. The quality of video in the standard video delivery case is significantly lower than the videos aggregated at different rates; aggregation is able to reduce the video size and adjust the encoding rates to fit the transported video within the available capacity.

C. CA in Asymmetric Topologies

Thus far, we evaluated symmetric topologies where routers at each level can be allocated similar bandwidth. We now



(a) Video Quality and Frame Success Rate (b) Throughput and Goodput (c) Network energy consumption

Fig. 8. Video Quality, Network Throughput and Energy Efficiency: Video transmission requires a low number of frame losses are minimum for good video quality. As frame error rate increases, video quality deteriorates, and energy efficiency drops.

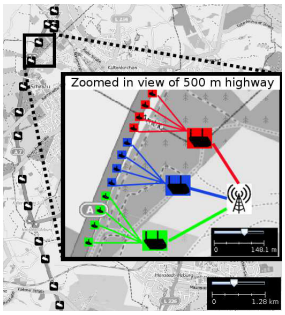


Fig. 9. Surveillance infrastructure on Highway-A7 in Germany. The inset shows proposed the topology for monitoring of a 500 m stretch of the highway.

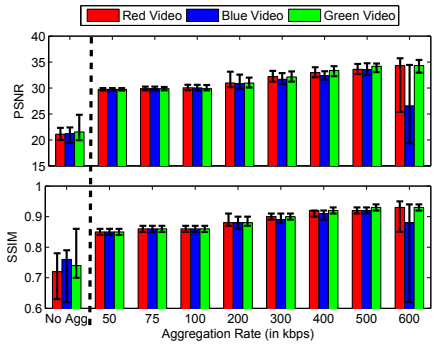


Fig. 10. Video quality with and without CA

examine larger asymmetric scenarios using a realistic border surveillance scenario with long-range WiMAX links.

Border Surveillance Scenario: Border surveillance requires monitoring large remote and inaccessible areas. Figure 4(a) shows the Nogales area in Arizona where the US border is monitored using an SCN [23]. Currently, there are seven fixed surveillance and communication towers [23]. Each surveillance tower covers a portion of the border area that is already identified, and transmits the real-time video to the OC. Such real-time video is vital for instantly detecting and reacting to illegal border breaches. The sparse deployment of cameras creates large coverage holes that are unmonitored, thus prone to illegal movement of people and smuggling of drugs.

In the scenario we consider, we propose deployment of cameras near the border with the support of WiMAX Base Stations (BS) to relay the video to the OC. WiMAX is suitable for such scenarios because of its longer range (approx. 2-3 km). We choose simulation parameters to match the current infrastructure [23]; each tower is 21 m high and transmits at 43 dBm power over a 7 GHz range. Using this configuration, we are able to achieve a link capacity of around 30 Mbps at 500 m, and 10 Mbps at 2.5 km range. We assume that each BS operates on a different frequency range.

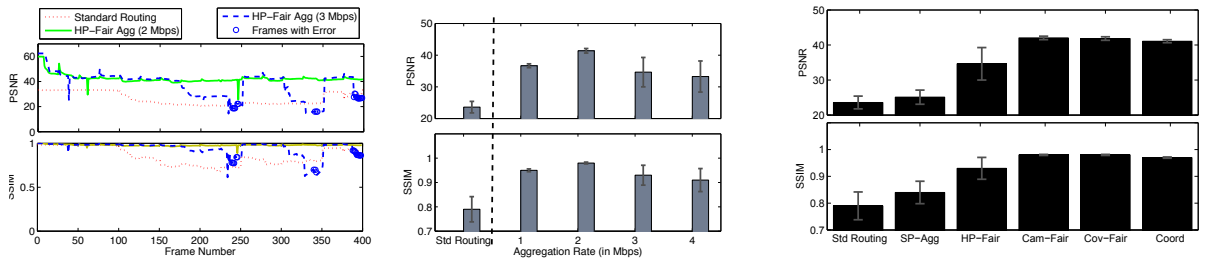
We simulate an example scenario where four cameras (C1 to C4) are randomly deployed in a strip of $8 \times 1\,\text{km}^2$, as shown in Figure 4(b). The inset of the figure shows the snapshot of the sample surveillance video that overlooks long border. Using projective geometry and camera parameters, such as focal length, we map the area seen by the camera to a part of the overall Field-of-View (FoV) of the surveillance area [24]. We cut the parts of the FoVs of the original video and scale all the videos to the same dimension. We then encode the videos at 2 Mbps, which are streamed in the simulator from the camera nodes.

Recall that in the HP-Fair scheme, each base-station (BS1-BS6 in Figure 4) aggregates at the same rate. Figures 11(a) and 11(b) compares the video quality between different schemes. We simulate four scenarios where all base-stations aggregate the video at 1,2,3 and 4 Mbps.

HP-Fair provides significant improvement. As the encoding rate increases towards network capacity, the video quality of HP-Fair increases. However, as the capacity exceeds *2 Mbps*, we observed substantial rates of frame drops (as seen by the *HP-Fair Agg 3 Mbps* curve in Figure 11(a)). These drops in turn cause lower and fluctuating measured PSNR and SSIM (Figure 11(b)).

Cam-Fair, Cov-Fair and CC schemes also exhibited behavior similar to HP-Fair, where the video quality increases as the encoding rate increased until a threshold value. However the drop in video quality occurred at different thresholds. We observed that Cam-Fair, Cov-Fair and CC were able to sustain a rate of 3 Mbps video at the OC; note that HP-Fair was saturated at 2 Mbps.

Figure 11(c) compares all the schemes when the collective



(a) Frame Level Quality (b) Summary Quality (c) Performance for output = 3 Mbps video at OC

Fig. 11. Video quality in HP-Fair and other schemes: HP-Fair and other schemes achieves good video quality when the aggregation rate is 3 Mbps. Among the different schemes, Cov-Fair and Cam-Fair perform the best.

consolidated video is delivered to the OC at 3 Mbps, which is the bit-rate that provides best video quality observed across all schemes. We also include a naïve scheme (Single-Point Aggregation or SP-Agg) where the first router where all videos meet (BS4 in Figure 4) is chosen as the only aggregation point. The CA schemes adjust their encoding rate at the aggregation points accordingly. CC, Cam-Fair and Cov-Fair perform the best across all the schemes.

VII. Related Work

CTCV combines video consolidation with coverage and bandwidth aware end-to-end video delivery for SCN environments. In this section, we discuss related work relative to specific aspects of the proposed framework.

Video stitching: Related to video consolidation, stitching of multiple videos from multiple cameras – with possible multiple perspectives – has been investigated at the OC to provide a combined view for human operators [25]–[28]. El-Saban *et al.* propose real-time stitching of videos streamed from different mobile phones on a centralized server [25]. Similarly, Akella *et. al.* stitch videos deployed in UAVs for a battlefield monitoring application centrally. Qin et al. [27] consider online panorama generation of videos transmitted by networked robotic cameras. Our work differs from these works in that mosaicing is carried out in the network to reduce the bandwidth requirements. Alternative models for consolidation corresponding to other video presentation requirements are also possible within the proposed framework.

Camera placement [29] and coverage works [30], [31] are related to CTCV in that they attempt to reduce overlap between cameras. For example, Yildiz et al. [29] present a camera placement scheme to generate panoramic video of a given area with minimum number of cameras. However, camera location planning is not possible for organically growing or mobile SCNs. In such situations, CTCV can complement planning to reduce the overhead from resulting coverage overlap.

Multimedia Wireless Networking: The second component of CTCV is coordinated bandwidth allocation. Several studies optimize video transmission on a network using scheduling, routing and video rate-control [10]–[12]. Video summarization is used to further reduce the required bandwidth by presenting a non-video summary, such as text and motion paths [32], [33]. These works focus on reducing the bandwidth usage of a single video stream and do not consider SCN scenarios where multiple cameras are concurrently streaming.

Other studies consider transmission of multiple videos over network. Liew *et al.* [34] propose joint encoding of multiple videos before transmitting them on the network. Video streaming of multiple streams in wireless mesh networks is considered by Navda *et al.* [35], where the focus is on improving the throughput by identifying and using spatially separated routes for video delivery. Our work is orthogonal: we propose a network-aware delivery model in SCNs rather than a routing or encoding protocol. Thus, we believe the above schemes can be used in conjunction with CTCV to further improve performance. CTCV can be considered as a generalization of video transport for 3D video [36], which consists of the transport of output from two correlated stereo cameras. CTCV generalizes this problem in several ways including the use of multiple cameras with arbitrary partial overlap between them.

Data aggregation in sensor networks: Aggregation in traditional sensor networks bears conceptual similarity to video consolidation in terms of selecting aggregation points and using appropriate aggregation functions [5]–[8]. Specifically, to aggregate data in sensor networks, an aggregation tree is constructed from sensors to the sink [7], in which all nodes except the leaf nodes can be the data aggregation points. Common aggregation functions used are simple operators such as MAX or AVERAGE [7], [37]. To our knowledge, this is the first work that considers aggregation of a complex data type such as video.

VIII. Conclusions

This paper introduces the problem of coordinated transport of correlated video (CTCV) present in Smart Camera Networks (SCN) deployed for wide-area surveillance. When events occur in such networks, often the observer is interested in viewing multiple videos from nearby cameras that overlap in coverage. Since video traffic has high bandwidth demands, naïve transfer of such videos often overwhelms the capacity of the network.

We modeled the CTCV problem as a joint routing and scheduling problem using the GNUM framework. We then

proposed two approaches for video consolidation. The first approach, consolidation by aggregation (CA), detects and blends areas of overlap in videos at intermediate routers. The second approach, consolidation by coordination (CC), further improves on CA by having the cameras avoid sending the redundant data to begin with. For bandwidth allocation, we investigated different policies for allocating the bandwidth among different routers in CA.

We evaluated CTCV schemes using public surveillance videos as well as video feeds from a small camera testbed. We showed that the coverage aware policies provide the significantly better performance (more than 10 dB gain in most scenarios) in realistic simulation scenarios based on real deployments of SCNs and public video feeds.

REFERENCES

[1] H. Aghajan and A. Cavallaro, *Multi-Camera Networks: Principles and Applications*. Academic Press, 2009.

[2] H. Collins, "Video Camera Networks Link Real-Time Partners in Crime-Solving," Feb 2012. [Online]. Available: http://www.govtech.com/public-safety/Video-Camera-Networks-Link-Real-Time-Partners-in-Crime-Solving.html

[3] K. Abas, C. Porto, and K. Obraczka, "Wireless Smart Camera Networks for the Surveillance of Public Spaces," *Computer*, vol. 47, no. 5, pp. 37–44, May 2014.

[4] N. G. L. Vigne, S. S. Lowry, J. A. Markman, and A. M. Dwyer, "Evaluating the Use of Public Surveillance Cameras for Crime Control and Prevention," Sep 2011.

[5] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 2, pp. 28–36, 2002.

[6] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: A survey," *Wireless Communications, IEEE*, vol. 14, no. 2, pp. 70 –87, april 2007.

[7] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: a Tiny AGgregation service for ad-hoc sensor networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 131–146, Dec. 2002.

[8] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *Networking, IEEE/ACM Transactions on*, vol. 11, no. 1, pp. 2 – 16, Feb 2003.

[9] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 255–312, Jan 2007.

[10] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod, "Cross-layer design of ad hoc networks for real-time video streaming," *Wireless Communications, IEEE*, vol. 12, no. 4, pp. 59 – 65, aug. 2005.

[11] S. Milani and G. Calvagno, "A low-complexity cross-layer optimization algorithm for video communication over wireless networks," *Trans. Multi.*, vol. 11, no. 5, pp. 810–821, 2009.

[12] M. van der Schaar and D. S. Turaga, "Cross-Layer Packetization and Retransmission Strategies for Delay-Sensitive Wireless Multimedia Transmission," *Multimedia, IEEE Transactions on*, vol. 9, no. 1, pp. 185 –197, jan. 2007.

[13] S. Winkler and P. Mohandas, "The Evolution of Video Quality Measurement: From PSNR to Hybrid Metrics," *Broadcasting, IEEE Transactions on*, vol. 54, no. 3, pp. 660 –668, sept. 2008.

[14] H. Dersch *et al.*, "Panorama tools," 1998–2008. [Online]. Available: http://panotools.sourceforge.net/

[15] D. Ding, F. Metze, S. Rawat, P. F. Schulam, S. Burger, E. Younessian, L. Bao, M. G. Christel, and A. Hauptmann, "Beyond audio and video retrieval: towards multimedia summarization," in *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, ser. ICMR '12. New York, NY, USA: ACM, 2012, pp. 2:1–2:8.

[16] Cisco, "Taking 3D to the Next Level," Feb 2011. [Online]. Available: http://newsroom.cisco.com/feature-content?type=webcontent&articleId=5910450

[17] "Enblend and enfuse." [Online]. Available: http://enblend.sourceforge.net/

[18] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, April 2006, pp. 1–13.

[19] "Qualnet network simulator," http://www.scalable-networks.com/.

[20] J. Klaue, B. Rathke, and A. Wolisz, "EvalVid: A Framework for Video Transmission and Quality Evaluation," in *Computer Performance Evaluation. Modelling Techniques and Tools*, ser. Lecture Notes in Computer Science, P. Kemper and W. Sanders, Eds. Springer Berlin / Heidelberg, 2003, vol. 2794, pp. 255–272.

[21] "Ffmpeg and libavcodec." [Online]. Available: http://ffmpeg.org/

[22] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600 –612, april 2004.

[23] Department of Homeland Security, "Integrated Fixed Towers." [Online]. Available: https://www.fbo.gov/?s=opportunity&mode=form&id=5d1e009b6009b6a503ddcb5e61af8258&tab=core&_cview=1

[24] R. Radke, "Multiview Geometry for Camera Networks," *Multi-camera networks: principles and applications*, 2009.

[25] M. A. El-Saban, M. Refaat, A. Kaheel, and A. Abdul-Hamid, "Stitching videos streamed by mobile phones in real-time," in *Proceedings of the 17th ACM international conference on Multimedia*, ser. MM '09. New York, NY, USA: ACM, 2009, pp. 1009–1010.

[26] P. Akella and A. Ganz, "Panorama - pervasive networking architecture for multimedia transmission in battlefield environments," in *Military Communications Conference, 2004. MILCOM 2004. 2004 IEEE*, vol. 2, Nov 2004, pp. 1107–1113.

[27] N. Qin and D. Song, "On-demand sharing of a high-resolution panorama video from networked robotic cameras," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 29 2007-nov. 2 2007, pp. 3113–3118.

[28] F. Perazzi, A. Sorkine-Hornung, H. Zimmer, P. Kaufmann, O. Wang, S. Watson, and M. Gross, "Panoramic video from unstructured camera arrays," *Computer Graphics Forum*, vol. 34, no. 2, pp. 57–68, 2015.

[29] E. Yildiz, K. Akkaya, E. Sisikoglu, M. Sir, and I. Guneydas, "Camera Deployment for Video Panorama Generation in Wireless Visual Sensor Networks," in *Multimedia (ISM), 2011 IEEE International Symposium on*, dec. 2011, pp. 595 –600.

[30] V. Munishwar and N. Abu-Ghazaleh, "Coverage Algorithms in Visual Sensor Networks," *ACM Transactions on Sensor Networks*, Jul. 2013.

[31] V. Munishwar, V. Kolar, and N. Abu-Ghazaleh, "Coverage in visual sensor networks with Pan-Tilt-Zoom cameras: The MaxFoV problem," in *INFOCOM, 2014 Proceedings IEEE*, April 2014, pp. 1492–1500.

[32] W.-C. Feng, E. Kaiser, W. C. Feng, and M. L. Baillif, "Panoptes: scalable low-power video sensor networking technologies," *ACM Trans. Multimedia Comp. Comm. Appl.*, vol. 1, no. 2, pp. 151–167, 2005.

[33] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 34, no. 3, pp. 334 –352, Aug 2004.

[34] S. Liew and C.-Y. Tse, "Video aggregation: adapting video traffic for transport over broadband networks by integrating data compression and statistical multiplexing," *Selected Areas in Communications, IEEE Journal on*, vol. 14, no. 6, pp. 1123 –1137, aug 1996.

[35] V. Navda, A. Kashyap, S. Ganguly, and R. Izmailov, "Real-time Video Stream Aggregation in Wireless Mesh Network," in *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, sept. 2006, pp. 1 –7.

[36] C. G. Gurler, B. Gorkemli, G. Saygili, and A. M. Tekalp, "Flexible transport of 3-d video over networks," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 694–707, April 2011.

[37] S. Lindsey, C. Raghavendra, and K. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 13, no. 9, Sep 2002.